200308761-1 (HPC.0018US)

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: REMAPPING ROUTING INFORMATION ENTRIES IN AN EXPANDER

INVENTORS: JOSEPH E. FOSTER,
11907 Moorcreek Drive
Houston, Texas 77070

ROBERT C. ELLIOTT,
13222 Champions Centre Drive
Houston, Texas 77069

HUBERT E. BRINKMANN, JR.,
6106 Darby Way
Spring, Texas 77389

and

JAMES R. REIF
15022 Rose Valley Drive
Houston, Texas 77070

REMAPPING ROUTING INFORMATION ENTRIES IN AN EXPANDER

BACKGROUND

[0001] In certain applications, such as in a network environment, relatively large amounts of data may have to be stored in storage subsystems of computer systems. In a network environment, many users store data and programs on one or more computer servers, which usually include or are attached to one or more storage subsystems of relatively large capacity. A computer server storage subsystem can be made up of a large number of storage devices, including hard disk drives, tape drives, compact disc (CD) drives, digital versatile disc (DVD) drives, and so forth.

[0002] A popular interface for coupling storage devices (and other peripheral devices) to a computer system is the small computer system interface (SCSI). A SCSI interface is traditionally a parallel interface (having multiple signals) to provide increased bandwidth in communications between a computer and a peripheral device. However, parallel interfaces may not be able to offer reliable performance at very high operating frequencies.

[0003] To address issues associated with traditional SCSI interfaces, a Serial Attached SCSI (SAS) Standard has been proposed. The SAS Standard defines the rules for exchanging information between SCSI devices using a serial interconnect. The SAS Standard also defines the rules for exchanging information between AT attached (ATA) host and ATA devices using the same serial interconnect. ATA is a standard for the internal attachment of storage devices to hosts. One version of the SAS Standard is defined by Working Draft American National Standard, "Information Technology-Serial Attached SCSI (SAS)," Revision 5, dated July 9, 2003.

[0004] One feature of a SAS system is that multiple SAS domains can be defined, with each domain having a tree of interconnected devices that include one or more expanders. An expander increases the number of interfaces available to couple to peripheral devices (such as storage devices) within a given SAS domain. Expanders can be coupled to other expanders to further expand the capacity to attach to additional peripheral devices. Usually, each SAS domain (or SAS expander tree) is associated with one or more initiators. An initiator

responds to commands from software in a computer system for accessing storage devices in a domain to retrieve data or to write data.

[0005] Each expander includes phys that are each coupled to an initiator, another expander, or a target device. A phy is a type of interface that communicates with another phy over a link. Phys are associated with route tables that contain routing information used to route an access request through phys of an expander such that the access request can reach the intended target device. The route table entries for certain types of phys are not utilized and thus are disabled. However, because the current version of the SAS Standard implements static route table binding, such unused route table entries are not available for use by other phys. As a result, the number of devices that a SAS expander can support is limited.

SUMMARY

[0006] In general, according to one embodiment, a system includes a first expander having plural interfaces to couple to at least one of a peripheral device, a controller, and another expander. The first expander has a storage to store entries containing routing information used to route a request received by the first expander to one of the interfaces, wherein each interface is allocated a respective set of the routing information entries. The system also includes mapping logic operable to remap unused routing information entries allocated to one of the interfaces to one or more other interfaces to expand capacity of the one or more other interfaces.

[0007] Other or alternative features will become apparent from the following description, from the drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Figs. 1A-1B are a block diagram of an example computer system incorporating an embodiment of the invention that includes Serial Attached Small Computer System Interface (SAS) storage devices.

[0009] Fig. 2 is a block diagram of components of an expander according to an embodiment in a SAS domain in the computer system of Fig. 1.

[0010] Fig. 3 is a schematic diagram of an example arrangement of an expander according to an embodiment containing multiple SAS phys that are associated with respective route table entries.

[0011] Fig. 4 illustrates mapping of unused and redundant route table entries of some of the SAS phys in the expander of Fig. 3 to other SAS phys in the expander, in accordance with an embodiment.

[0012] Fig. 5 is a block diagram of mapping logic according to an embodiment for mapping route table entries.

DETAILED DESCRIPTION

[0013] Referring to Figs. 1A-1B, a computer system 107 according to one example arrangement includes a central processing unit (CPU) 101, memory 130, and a bridge device such as north bridge 120. The north bridge 120 may be coupled through a bus 140 to another bridge device such as south bridge 180. South bridge 180 may be coupled to various devices, including a non-volatile memory 185.

[0014] Additionally, the north bridge 120 may be coupled to an input/output (I/O) bridge 191 through an I/O bus 145. The I/O bridge 191 is in turn coupled to several peripheral devices, such as a network interface card (NIC) 196, and a SAS (Serial Attached Small Computer System Interface) controller 105 (Fig. 1B).

[0015] The SAS controller 105 is part of a SAS I/O subsystem (identified by numeral 100 in Fig. 1B). The SAS I/O subsystem 100 has an architecture that conforms with the SAS Standard, with one version described in Working Draft American National Standard, "Information Technology-Serial Attached SCSI (SAS)," Revision 5, dated July 9, 2003. The SAS Standard defines the rules to enable the exchange of information between SCSI (small computer system interface) devices over a serial interconnect. SCSI devices include storage devices such as hard disk drives, compact disc (CD) drives, digital versatile disc (DVD) drives, and other mass storage devices. In other embodiments, SCSI devices can also include other types of peripheral devices.

[0016] Read or write operations to storage devices in the SAS I/O subsystem 100 may be generated by the CPU 101. In response to such read or write requests, the SAS controller 105 initiates read or write operations to the storage devices in one or more of first storage tree 120, second storage tree 160, third storage tree 170, and fourth storage tree 180 using SAS physical interconnections and messaging defined by the SAS Standard. In other arrangements, additional SAS controller(s) can also be present in the system.

[0017] In one embodiment, the SAS controller 105 is implemented as an application-specific integrated circuit (ASIC) that includes firmware. In other embodiments, the SCSI controller 105 can be implemented with other types of devices, such as processors, microcontrollers, and so forth. The SAS controller 105 is coupled to an expander 110 through links 106a-106d, according to one example. An expander is an input/output control device such as a switch that receives information packets at a port from a source and routes the information packets through a selected one of plural other ports to the correct destination.

[0018] Each end of a link 106 couples to an interface within each of the SAS controller 105 and expander 110. In one embodiment, such an interface includes a physical device referred to as a "phy" (PHYsical device) as defined by the SAS Standard. A phy includes a transceiver to electrically communicate over the link 106 with a transceiver in another phy. According to SAS, each link is full duplex, such that information can be transferred simultaneously in both directions over the link. Each link 106 is a receive differential pair and a transmit differential pair.

[0019] In the example arrangement shown, the expander 110 is coupled over links to devices in multiple storage trees 120, 160, 170, and 180. The links between the expander 110 and the storage tree 120 are labeled 116a and 116b. The storage tree 120 includes three additional expanders 125, 130, and 135. The expander 125 is connected to the expander 110, storage devices (SD) SDA, SDB, SDC, SDX, SDY, and SDZ, expander 130, and expander 135. At the lowest level of the storage tree 120, expander 130 and expander 135 are each further connected to multiple storage devices. Each of expanders 110, 125, 130 and 135 includes a routing controller (described in greater detail below) that allows information received by one port to be transmitted to an expander or storage device through another port in the expander.

[0020] Turning to Fig. 2, an example arrangement of the components of one of the expanders 110, 125, 130, and 135 in Fig. 1B are described in further detail. The expander includes storage to store routing tables (each labeled "TABLE") for respective phys (each labeled "PHY"). A routing table 217 for one of the phys includes expander route entries 230a, 230b, ..., 230n, each of which may include an enable/disable bit 220 and a SAS address 225. A SAS address is a unique identifier assigned to an initiator, expander, or storage device. The routing table for each phy may include up to 12 route entries, according to one example implementation. A routing controller 240 in the expander 110 is able to access each routing table to allocate and remap the route entries in each of the routing tables as desired, as described further below.

[0021] Enable/disable bit 220 in a route entry indicates whether the route entry contains a valid SAS address. In some configurations, not all route entries in a routing table may be utilized. The enable/disable bit 220 for an un-utilized route table entry is set to the disable state.

[0022] Each phy in an expander has a routing attribute to indicate a routing capability of a phy. There are several types of routing attributes: direct routing attribute, subtractive routing attribute, and table routing attribute. A phy with the direct routing attribute indicates that the phy may be used to route a read/write request to an end device (e.g., a storage device or a host). A phy with the table routing attribute indicates that the phy may be used to route a read/write request using a routing table. A phy with the subtractive routing attribute indicates that the phy is used to route unresolved read/write requests (that is, requests not routed to a phy with a direct routing attribute or to a phy with a table routing attribute).

[0023] The routing attribute in combination with the type of device the phy is connected to indicate the routing method used by the phy. A phy with a direct routing attribute connected to an end device means that the direct routing method is used. A phy with the direct routing attribute that is connected to an expander also means that the direct routing method is used. The routing table for a phy connected by the direct routing method does not contain any valid route table entries and thus the enable/disable bit 220 is disabled for each route entry.

[0024] A phy with a subtractive routing attribute has the capability of functioning as an input phy in the expander (subtractive phys are upstream of table phys). A phy with the

subtractive routing attribute connected to an expander means that the phy is connected according to the subtractive routing method. On the other hand, a phy with the subtractive routing attribute connected to an end device means that the phy is connected according to the direct routing method. The routing table for a phy connected according to the subtractive routing method does not contain any valid route table entries and thus the enable/disable bit 220 is disabled for each route entry.

[0025] A phy with a table routing attribute indicates that the phy can function as an interface to another expander. A phy with the table routing attribute connected to an expander means that the phy is connected according to the table routing method. However, a phy with the table routing attribute connected to an end device means that the phy is connected according to the direct routing method. The routing table for a phy connected according to the table routing method may include valid route table entries used by the routing controller to route read/write requests and perform information transfers.

[0026] The table below summarizes the routing method used based on the phy attribute and type of device connected to the phy:

Phy Attribute	Connected Device	Routing Method Used
Direct	End Device	Direct
Direct	Expander	Direct
Subtractive	End Device	Direct
Subtractive	Expander	Subtractive
Table	End Device	Direct
Table	Expander	Table

[0027] Fig. 3 illustrates an expander 300 (which can be any of one of the expanders 125, 130, and 135 shown in Fig. 1B) with unused and redundant route table entries. The expander 300 is connected over a wide port 302 to another expander 304. A port in an expander includes one or plural phys. A "wide port" includes plural phys. In the example shown, the wide port 302 includes four phys 304a-304d. The phys 304a-304d are connected according to the subtractive routing method, and thus route table entries (A, B, C, D) allocated to the respective subtractive routing phys 304a-304d are not used. Similarly, the port 306, which is directly connected to an end device 307 (initiator or target), is allocated route table entries (E)

that are unused. Phy 308 is not connected to any device, and thus route table entries (F) allocated to the phy 308 are also unused.

[0028] The expander 300 is connected to another expander 312 through a wide port 310 having phys 314a-314d. The phys 314a-314d are connected according to the table routing method. However, since the phys 314a-314d are all part of the same wide port 310, the route table entries (G, H, I, J) for the phys 314a-314d, respectively, are redundant.

[0029] According to a current version of the SAS Standard, which implements static route table binding, the unused or redundant route table entries are not allocated to other phys, which reduces the number of SAS devices to which the expander can be connected. To increase the number of SAS devices that the expander 300 can be connected to, a dynamic route table binding scheme according to some embodiments of the invention is implemented that employs (1) remapping of unused route table entries from one port to another port(s), and (2) aliasing of redundant route table entries of a wide port so that each phy of the wide port uses the same route table entries, leaving the remaining route table entries for use by other port(s).

[0030] As shown in Fig, 4, after remapping has been performed, the following unused route table entries are remapped to the phy 314a in the wide port 310: route table entries A, B, C, D originally associated with respective subtractive routing phys 304a, 304b, 304c, 304d; the route table entries E originally associated with the direct routing port 306; and route table entries F originally associated with the unused port 308. Also, aliasing is performed to map the route table entries H, I, J (which would be redundant of G without aliasing) to the phy 314a. After the remapping and aliasing, route table entries A-J are all mapped to phy 314a. Note that the other phys 314b, 314c, 314d of the wide port 310 (Fig. 3) all use the same table entries A-J.

[0031] If the number of route table entries per phy is N, then the total number of route table entries that are available for each phy 314a-314d is 10xN, rather than the 4xN route table entries available to phys 314a-314d before remapping and aliasing. As a result, the number of SAS devices in the expander tree that the expander 300 supports is increased substantially.

[0032] Effectively, the route table entries of the expander make up a virtual route table (rather than plural route tables dedicated to respective phys). The entries of the virtual route table can be dynamically mapped to phys that actually use the route table entries. Thus, as explained above, unused route table entries of one phy are remapped to one or more other phys, and redundant route table entries are aliased such that plural phys of the same SAS port share the same set of route table entries to avoid such redundancy.

[0033] Fig. 5 depicts the logic in an expander used to perform remapping and aliasing according to some embodiments. In response to SMP (Serial Management Protocol) commands from discovery software 350 (executable in the computer system 107 of Fig. 1), a route table entry mapping (RTEM) logic 352 populates the route table entries in the expander 300 (Figs. 3-4). SMP is the protocol used by SAS devices to communicate management information with other SAS devices in a SAS domain. The RTEM logic 352 is located in each expander in a SAS domain to enable remapping and aliasing of unused and redundant route table entries.

[0034] According to one embodiment, remapping of route table entries can be performed without modification of the discovery software 350. To perform the remapping and aliasing in this embodiment, the RTEM logic 352 accesses configuration control information stored in a non-volatile memory 354 (such as flash memory or electrically erasable and programmable read-only memory). The configuration control information can be pre-installed in the non-volatile memory 354 based upon the configuration of the SAS domain. A Routing Table SMP read/write request (for populating route table entries) from the discovery software 350 in the computer system is mapped by the RTEM logic 352 to route table entries 356 (which make up the virtual route table of the expander 300) based on the configuration control information in the non-volatile memory 354. The dynamic mapping or routing mechanism described above allows any number (zero or more) of route table entries to be assigned to a specific phy or a group of phys. The remapping or aliasing of route table entries 356 is determined by the content of the configuration control information stored in memory 354. In other words, the association of a route table entry with a particular phy is determined by the configuration control information.

[0035] In addition, steering logic 358 is used to ensure that the status due to an expander connection request is sent to the appropriate phy grant module or modules. A connection request is sent by an initiator to a target device to establish a connection between the initiator and the target device. A connection is established to enable the initiator to send commands, functions, and data to the target device. The target device specified in the connection request is matched to a route table entry in the route table 356. A match is indicated to the steering logic 358, which provides the routing status of the connection request to the appropriate one of the phys. The association of the matching route table entry to a given phy is based on the configuration control information stored in the memory 354. The routing status is sent to a grant module associated with the selected phy. Each phy is associated with a grant module to enable the phy to select one of multiple requests to process based on a predefined arbitration algorithm.

[0036] The mapping logic discussed above for performing remapping and aliasing can be performed by hardware, firmware, or software, or any combination of the above. Firmware and software are executable on a microcontroller, microprocessor, or other control unit. As used here, a "controller" refers to hardware, firmware, software, or a combination thereof. A "controller" can refer to a single component or to plural components (whether software, firmware, or hardware).

[0037] Data and instructions of the software and firmware are stored on one or more machine-readable storage media. The storage media include different forms of memory including semiconductor memory devices such as dynamic or static random access memories (DRAMs or SRAMs), erasable and programmable read-only memories (EPROMs), electrically erasable and programmable read-only memories (EEPROMs) and flash memories; magnetic disks such as fixed, floppy and removable disks; other magnetic media including tape; and optical media such as compact disks (CDs) or digital video disks (DVDs).

[0038] While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations there from. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.